EIC system user manual

how to use system

Feb 28th 2013

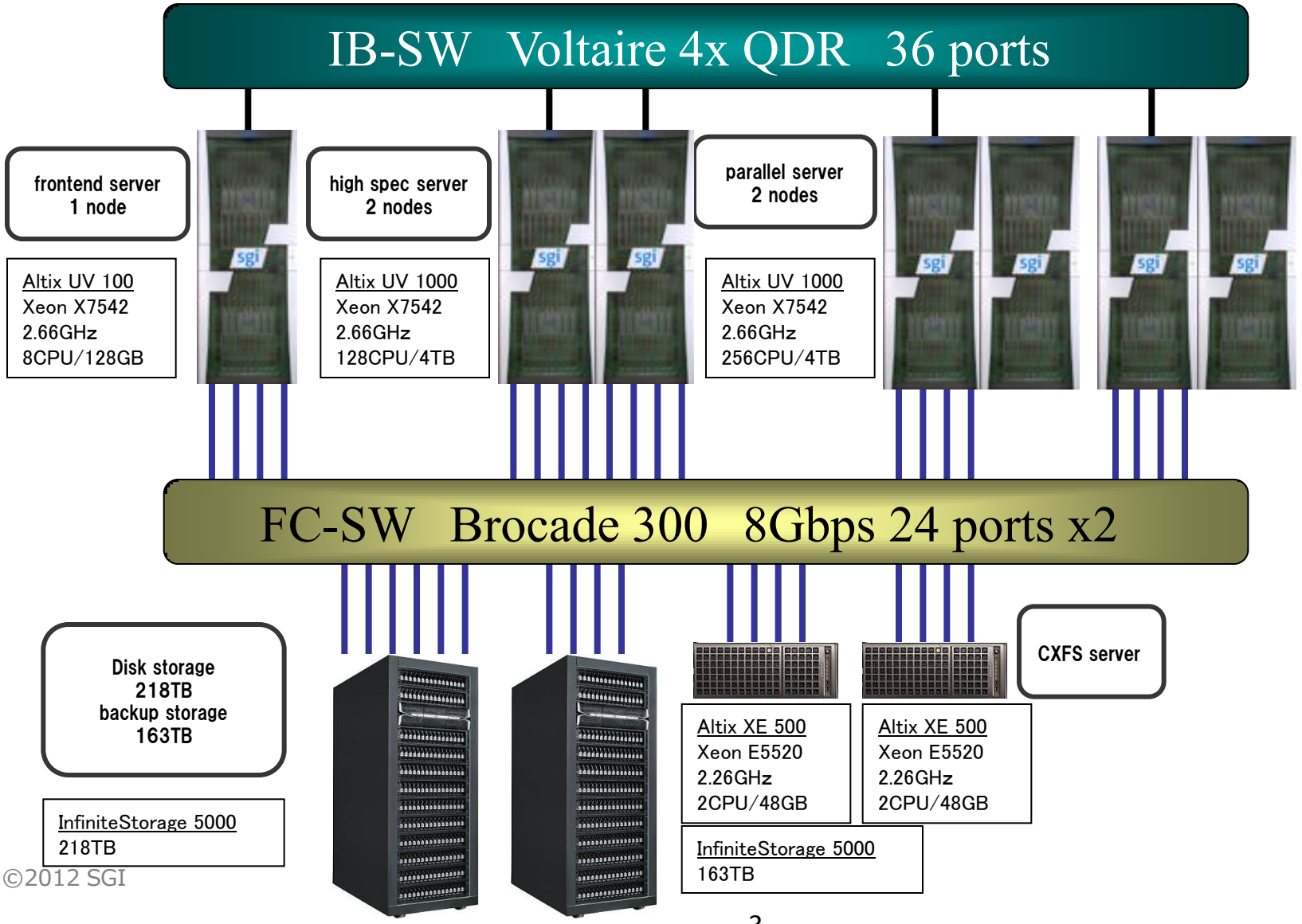
SGI Japan Ltd.

Index

- EIC system overview
- File system, Network
- User environment
- job script
- Submitting job
- Displaying status of job
- Canceling ,deleting job
- How to use peripheral devices
- Application software

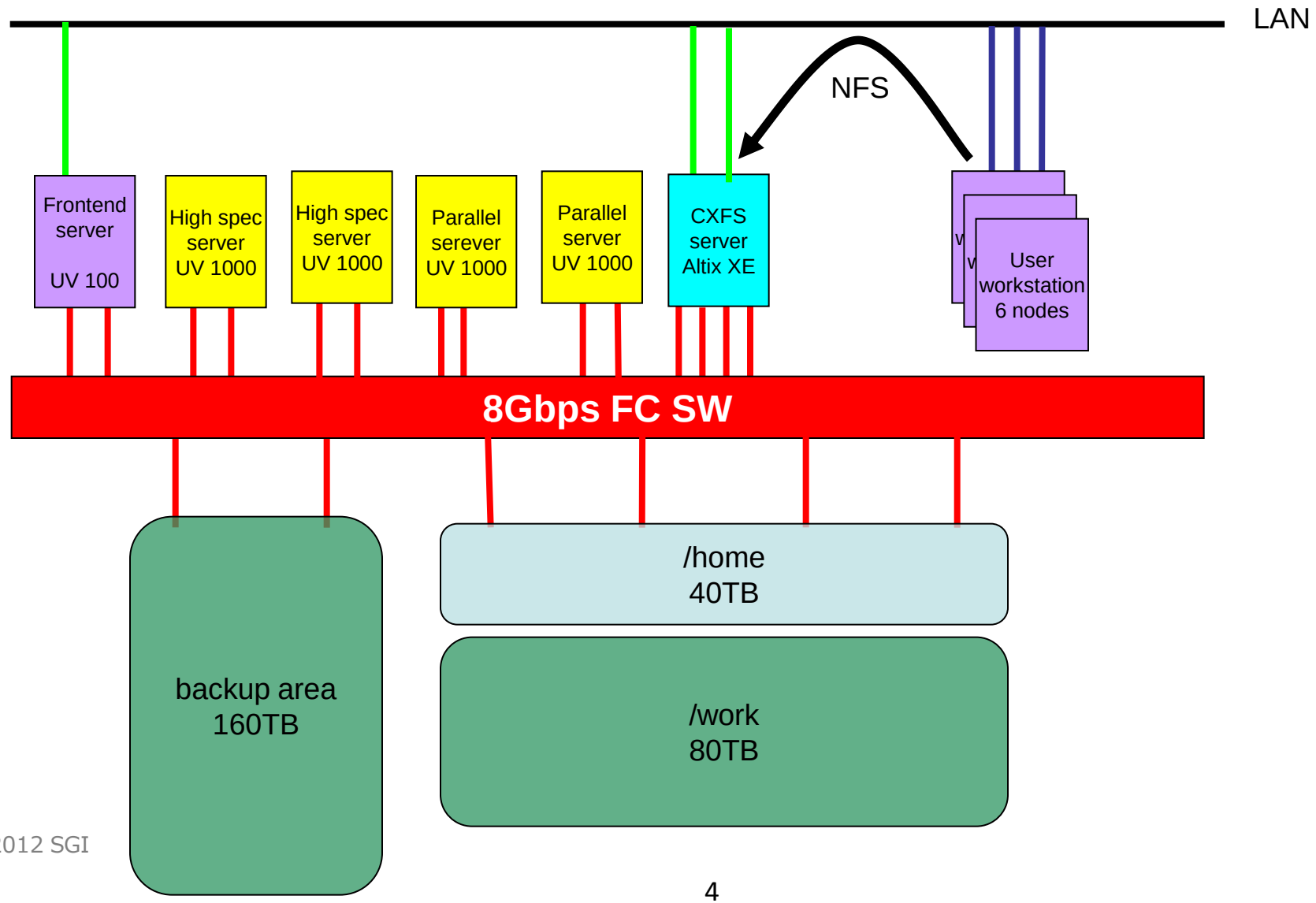
©2012 SGI

EIC system overview



©2012 SGI

File system, network



©2012 SGI

User environment

- EIC user can login and use the following servers.

hostname	Hardware	IP address	notice
eic	SGI UV 100	133.11.57.80	Frontend server
eic00	Dell Precision T3500	133.11.57.84	Workstation with DAT drive
eic01	Dell Precision T3500	133.11.57.85	Workstation with DAT drive
eic02	Dell Precision T3500	133.11.57.86	Workstation with Blu-ray drive
eic03	Dell Precision T3500	133.11.57.87	Workstation with Blu-ray drive
eic04	Dell Precision T3500	133.11.57.88	Workstation
eic05	Dell Precision T3500	133.11.57.89	Workstation

©2012 SGI

User environment

File system

- /home(home area)

  is total 40TB, you can use 150GB as default quota limit.

- /work(temporary area)

  is total 80TB, you can use 2000GB as default quota limit.

※ files which has never been accessed for 30 days are deleted.

©2012 SGI

User environment

TSS（interactive job） limitation

	limitation
CPU TIME	1 hour
MEMORY SIZE	1GB
STACK SIZE	4GB
CORE SIZE	0GB
Number of CPU	1

※Please use LSF batch software if you will run a job over TSS limitation.

©2012 SGI

User environment

- environment variable
  - On EIC system your environment variable was already set to use.
  - You don’t have to set environment variable by yourself.
  - There might be a trouble if you move environment variable files (ex .cshrc) in EIC system from other system, please pay your attention.
  - When you face any problem (ex can’t submit batch job, can’t check output file) after you migrated any environment variable files, delete your .cshrc on your home directory.

©2012 SGI

How to login

- How to login
  - Please login a frontend server “eic” for making a program, compiling, interactive debugging, submitting a batch job, frontend sever’s hostname is

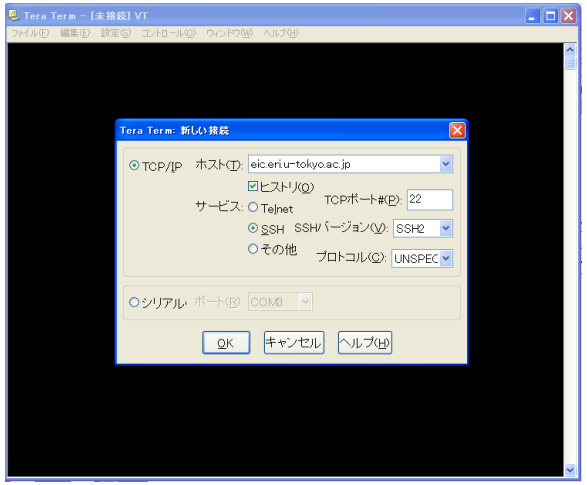
    eic.eri.u-tokyo.ac.jp
  - telnet,rsh,rlogin are not permitted on frontend server, please use ssh(SecureShell).
- Login from Linux workstation.
  - You can login to use SSH from your Linux workstation.

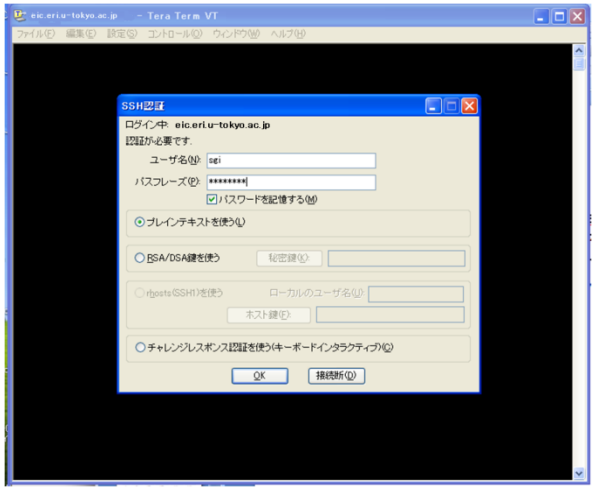
```
$ ssh –l username eic.eri.u-tokyo.ac.jp
```

©2012 SGI

How to login(Windows）

- how to login
  - Please use Windows' SSH software (TeraTerm,Putty)
    - HOST:eic.eri.u-tokyo.ac.jp
    - username:username
    - password:password
  - following is TeraTerm sample.





©2012 SGI

job script

Creating job script file

- Create job script file to submit a batch job, sample is in a right square.
- You must define ..
  - #BSUB-q “queue name”
  - #BSUB-n “number of cpu cores”
  - #BSUB -o “output file name”
  - dplace insert before command or program to improve performance.
- Attention
  - job ‘s standard output or error are temporary saved on /home, finally written into a file which you define as –o.
  - If you don’t define –o, outputs are sent to eic as email.
  - Email size is limited in 1MB, please define –o filename or re-direct filename on command line.

```
#!/usr/bin/csh
#BSUB -q A
#BSUB -n 1
#BSUB -o sample.out
dplace ./sample 4000
```

©2012 SGI

Submitting job

- Use “bsub” to submit a batch job.
  - Re-direct a job script to bsub command.

```
$ cat sample.csh
#!/usr/bin/csh
#BSUB -q A
#BSUB -n 1
#BSUB -o sample.out
dplace ./sample 4000
$ bsub < sample.csh
set LSB_SUB_MAX_NUM_PROCESSORS is 6
Job <958> is submitted to queue <A>.
```

job ID is printed.

©2012 SGI

Displaying job status

bstatus

- bstatus displays status of jobs you submitted.

```
$ bstatus
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
959     sgi     RUN   C          eic         24*eicp1    *para2.csh Feb  3 16:26
961     sgi     PEND  C          eic                     *para3.csh Feb  3 16:32
```

row “STAT” displays status of jobs.

RUN ----- job is running

PEND---- job is pending

bjobs command displays only your jobs.

©2012 SGI

Canceling, Deleting job

bkill

- bkill can cancel or delete jobs.
- define job ID.

```
$ bjobs
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
957     sgi     RUN   C          eic         24*eicp1    ./para.csh Feb  3 15:44

$ bkill 957
Job <957> is being terminated

$ bjobs
No unfinished job found
```

©2012 SGI

Queue configuration

Queue name	Runtime	Memory limit	Maximum memory limit	Parallel limit（cores）	Job limit（cores）
A	2h(cputime)	8GB	16GB	1(6)	1(6)
B	100h	32GB	32GB	1(6)	4(24)
C	80h	128GB	128GB	4(24)	3(72)
D	70h	256GB	256GB	8(48)	3(144)
E	50h	256GB	512GB	16(96)	2(192)
F	40h	512GB	1024GB	32(192)	1(192)
M	12h	8GB	8GB	MATLAB	

Queue name: Queue name
Runtime: Wallclock time limitation per job（only queue A limits cputime）
Memory limit: Memory limitation per job(Default)
Maximum memory limit: Memory limitation per job(If you define "–M" when you submit)
Parallel limit: number of CPU(cores) per job
Job limit: number of running job per user

©2012 SGI

MPI job script sample

```
$ cat go.24
#!/usr/bin/csh
#BSUB -q C
#BSUB -n 24
#BSUB -o test.out
mpirun -np 24 dplace -s1 ./xhpl < /dev/null >& out.mpi
```

- 24 mpi parallel job(sample)
- Define 24(number of parallel) on “BSUB –n” and “mpirun -np”
- Insert “dplace -s1” before running module name.

©2012 SGI

Submitting MPI job

```
$  bsub < ./go.24
set LSB_SUB_MAX_NUM_PROCESSORS is 24
Job <751> is submitted to queue <C>.

$ bjobs
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME    SUBMIT_TIME
751     sgi     RUN   C          eic         24*eicp1    ./go.36     Feb  3 10:13
```

- 24 parallel job runs on 4CPU（24cores）
- EIC servers have 12cores on local memory, all jobs are automatically set to multiple of 12 if number of cores is not multiple of 12.

©2012 SGI

OpenMP job script sample

```
$ cat para.csh
#!/usr/bin/csh
#BSUB -q D
#BSUB -n 48
#BSUB -o test.out
setenv OMP_NUM_THREADS 48
dplace  -x2 ./para < /dev/null >& out.para
```

- 48 OpenMP parallel job(sample)
- Define 48(number of parallel) on “BSUB –n” and environment variable ”OMP_NUM_THREADS”
- Insert “dplace -x2” before module name（-x2 is not required if build by GNU compiler.)

©2012 SGI

Submitting OpenMP job

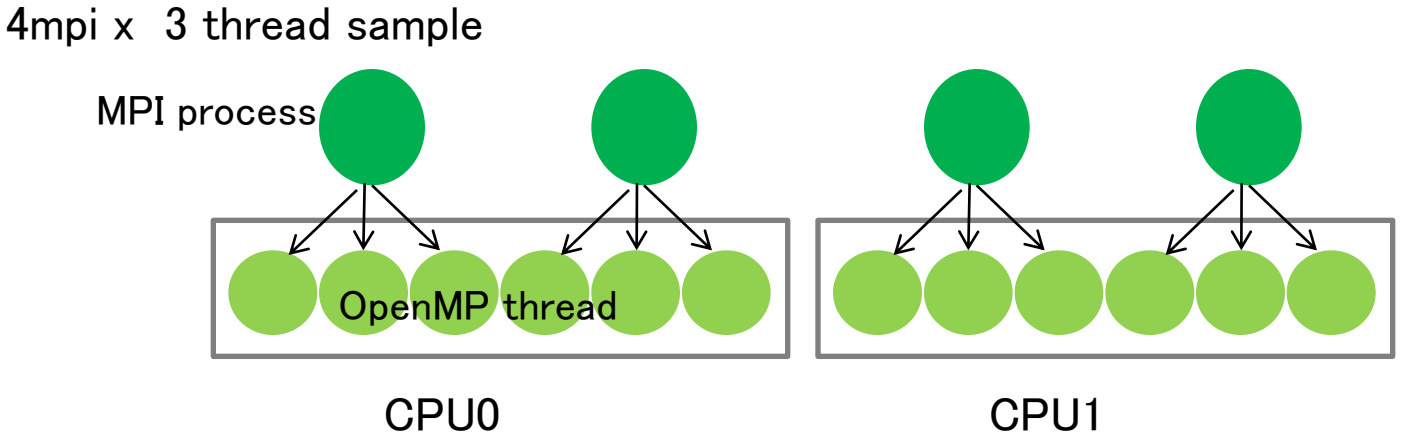
```
$  bsub < ./para.csh
set LSB_SUB_MAX_NUM_PROCESSORS is 48
Job <957> is submitted to queue <D>.

$ bjobs
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME    SUBMIT_TIME
957     sgi     RUN   D          eic         48*eicp1    ./para.csh Feb  3 10:13
```

- 48 parallel job runs on 8CPU（48cores）
- EIC servers have 12cores on local memory, all jobs are automatically set to multiple of 12 if number of cores is not multiple of 12.

©2012 SGI

MPI+OpenMP Hybrid Parallel job

- What is hybrid parallel ?
  - MPI processes boot OpenMP threads.
  - recommends 2,3, or 6 as number of OpenMP threads because OpenMP theads in same MPI process should use same local memory.



©2012 SGI

MPI+OpenMP Hybrid job script(sample)

```
$ cat go.csh
#!/bin/csh -x
#BSUB -q C
#BSUB -n 24
#BSUB -o hy1.out
limit stacksize unlimited
set np=8
set th=3
setenv OMP_NUM_THREADS ${th}
mpirun -np ${np} omplace -nt ${th} -c 0-23:bs=${th}+st=3 ./a.out
```

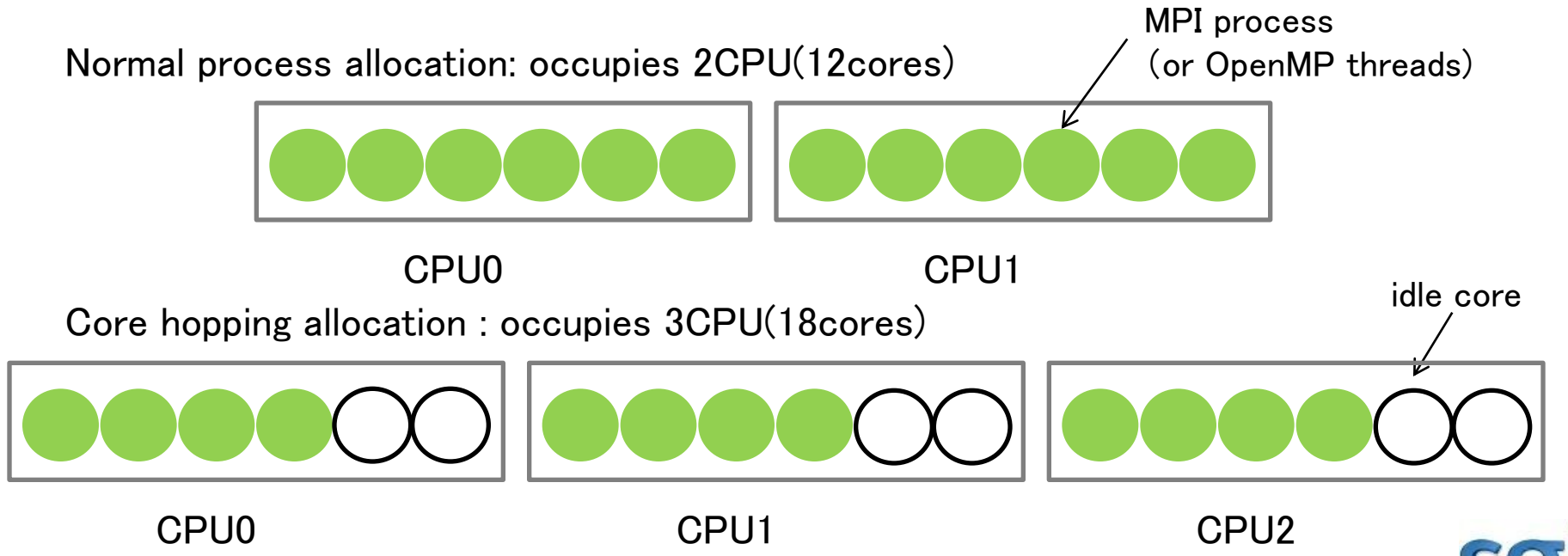
- 24 hybrid parallel job(8MPI x 3 threads)
- Define number of MPI for np, number of threads per MPI for th.
- Use omplace instead of dplace.
- insert following before command name.

  omplace -nt ${th} -c 0-23:bs=${th}+st=3
- -c means using 3cores from core 0 to core 23.

©2012 SGI

Core Hopping

- What is core hopping?
  - Normally job occupies all 6cores and local memory on same CPU socket.
  - You can reduce used cores per CPU if you would like to use wider memory band width per thread, it's called "core hopping".



©2012 SGI

Queue option for core hopping

- Define not only normal “-n” , but also “-P” how many cores you use per CPU.（define –P from 1 to 6)
- You have to select larger queue because you occupy more cores than number of –n. see following table.

Number of parallel	cores per CPU	Queue name	Queue Option	Number of occupied CPU（cores）
8	4	C	**#BSUB- q C** **#BSUB -n 8** **#BSUB -P 4**	2 (12)
32	4	D	**#BSUB- q D** **#BSUB -n 32** **#BSUB -P 4**	8 (48)
64	4	E	**#BSUB- q E** **#BSUB -n 64** **#BSUB -P 4**	16 (96)

©2012 SGI

Core hopping MPI job script

```
#!/usr/bin/csh
#BSUB -q D
#BSUB -n 32
#BSUB -P 4
#BSUB -o mpi4x8.out
source /opt/lsf/local/mpienv.csh 32 4
mpirun -np 32 ./xhpl < /dev/null >& out
```

32 parallel MPI job(4 cores per CPU)

#BSUB -n number of parallel

#BSUB -P cores per cpu（1~6）

source /opt/lsf/local/mpienv.csh [number of parallel] [cores per cpu]

（if you use sh(bash). /opt/lsf/local/mpienv.sh [number of parallel] [cores per cpu] ）

mpirun -np number of parallel command name…

Delete “dplace”

©2012 SGI

Core hopping OpenMP job script

```
#!/usr/bin/csh
#BSUB -q D
#BSUB -n 32
#BSUB -P 4
#BSUB -o out
set th=32
setenv OMP_NUM_THREADS ${th}
dplace -x2 0-3,6-9,12-15,18-21,24-27,30-33,36-39,42-45  ./para  >& out.para
または
omplace -nt ${th} -c 0-:bs=4+st=6 ./para >& out.para
```

32 parallel OpenMP job(4 cores per CPU)

#BSUB -n number of parallel

#BSUB -P cores per cpu（1~6）

omplace -nt [numberof parallel] -c 0-:bs=[cores per cpu] +st=6 [command name]….

©2012 SGI

Core hopping hybrid job script

```
#!/bin/csh
#BSUB -q C
#BSUB -n 32
#BSUB -P 4
#BSUB -o hy32-4.out
set np=8
set th=4
setenv OMP_NUM_THREADS ${th}
mpirun -np ${np} omplace -nt ${th} -c 0-:bs=${th}+st=6 ./a.out
```

BSUB -n number of parallel

#BSUB -P cores per cpu（1~6）

setenv OMP_NUM_THREADS [number of OpenMP threads]

mpirun -np [number of MPI] omplace -nt [number of OpenMP] -c 0-:bs=（core per cpu）+st=6 command….

©2012 SGI

Displaying core hopping job

qstatus

- qstatus displays core hopping job or normal job.

$ bjobs

```
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
33359   sgi     RUN   E          eic         48*eicp1    *t.sample5 Apr 27 11:28
```

$ qstatus

```
                                                c/p  CPUTIME/      CPUTIME   WALLTIME  MEMORY
JOB_ID    USER_NAME STAT Q  HOST PROC (-P)  WALLTIME   (hh:mm:ss) (hh:mm:ss)    (GB)
-------------------------------------------------------------------------------------------
33359          sgi  RUN E eicp1   48  4/6      29.4     00:50:08   00:01:42    17.0
```

cores per CPU

©2012 SGI

How to use printer

- how to print from eic, eicxx

Use “lpr”

eic%lpr -Pprinter_name PSfile_name

ex) eic%lpr –Pxdp1-6f test.ps

- how to print text file

Use “a2ps” to print text file

eic% a2ps -Pprinter_name ascii.txt

ex) eic%a2ps –Pxdp1-6f /home/sgi/ascii.txt

- Displaying print status

Use “lpq”

eic%lpq -Pprinter_name

ex) eic%lpq –Pxdp1-6f

Rank	Owner	Job	Files
1st	root	2	/home/sgi/test.f

- Cancel printing

Use “lprm”

eic%lprm -Pprinter_name request_ID

confirm request_ID “lpq -Pprinter_name”

ex)eic%lpq –Pxdp1-6f

Rank	Owner	Job	Files
1st	root	2	/home/sgi/test.f

eic%lprm –Pxdp6-1f 2

©2012 SGI

How to use DAT drive

- DAT drive
  - connect to eic00 and eic01.
    - /dev/st0------rewinding
    - /dev/nst0----no rewinding
  - Use tar or cpio
  - “mt” command to rewind or forward.
  - mt change uncompress or compress.
  - When you use DAT tape, please confirm compression mode.

©2012 SGI

How to use DAT

- Writing tape

$ mt -f /dev/st0 rewind	rewinding tape media
$ mt -f /dev/st0 compression 0	define 0 for uncompression, 1for compression.
$ cd /home/sgi/test	moving backup directory
$ tar cvf /dev/st0 .	writing current directory to tape, and rewind when it finishes

- Reading tape

$ mt -f /dev/st0 rewind	rewinding tape media
$ cd /home/sgi/test	moving writing directory
$tar xvf /dev/st0	writing data to current directory , and rewind when it finishes

- Confirming tape media

$ mt -f /dev/st0 rewind	rewinding tape media
$tar tvf /dev/st0	confirming tape media

- See online manual “man mt” or “man tar”

©2012 SGI

How to use Blu-ray drive

- Blu-ray drive
  - connect to eic02 and eic03
  - “bdr” command boots GUI writing software.

    $ bdr
  - confirm target drive as

    PIONEER BD-RW BDR-205 Rev1.08(p:1 t:0)
  - Select cursor menu on right side of drive name.
  - See User Manual Chapter4, manual is available from
  - http://wwweic.eri.u-tokyo.ac.jp/computer/manual/altixuv/doc/misc/bdrgui.pdf

©2012 SGI

Application software

AVS

- is available on workstations（eic00~eic05）.
- login to workstation, use “express” command.

  eic00$ express
- See manual
- http://kgt.cybernet.co.jp/article/2497/index.html

IMSL Fortran Library

- IMSL Fortran library Ver7.0 is available on EIC.
- TSS, OpenMP

  ifort –o [module name] $FFLAGS [source name] $LINK_FNL
- MPI

  ifort –o [module name] $FFLAGS [source name] $LINK_MPI

©2012 SGI

MATLAB

MATLAB

- you can use matlab on eic

Login to eic

% **ssh -X username@eic.eri.u-tokyo.ac.jp**

run matlab

% **matlab**

- You have to use LSF batch if you will run matlab over TSS limitation.
- See next page for matlab via LSF

	limitation
CPU TIME	1 hour
MEMORY SIZE	1GB
STACK SIZE	4GB
CORE SIZE	0GB
Number of CPU	1

©2012 SGI

MATLAB

MATLAB via LSF

- how to use matlab via LSF(batch)

Login to eic

% **ssh -X username@eic.eri.u-tokyo.ac.jp**

submit a job

% **bsub -q M –n 1 -Is /bin/tcsh**

or

% **bsub -q M –n 1 -Is /bin/bash**

Job <1519> is submitted to queue <M>.

<<Waiting for dispatch ...>>

<<Starting on eic>>

Confirm DISPLAY variable

% **env |grep DISPLAY**

eic:xx.0

Change DISPLAY variable.

% **setenv DISPLAY localhost:xx.0**

or

% **export DISPLAY=localhost:xx.0**

% **xhost +**

Run MATLAB

% **matlab**

©2012 SGI

Attention

- When you finish MATLAB, you have to
  - % exit
  - if you don’t exit, MATLAB license will be still used, other user will not be able to use it.
- MATLAB licenses are 10, you can’t run when all licenses are used.
  - when you bsub, “MATLAB License is over now” is displayed.
- You can use MATLAB on workstations（eic00~eic05）.
  - % matlab
  - （you can’t run when all licenses are used.)

©2012 SGI

©2012 SGI